

Review-Level Sentiment Classification with Sentence-Level Polarity Correction

Sylvester Olubolu Orimaye
(School of Information Technology,
Monash University Malaysia
sylvester.orimaye@monash.edu)

Saadat M. Alhashmi
(Department of Management Information Systems,
University of Sharjah,
Sharjah, United Arab Emirates
salhashmi@sharjah.ac.ae)

Eu-Gene Siew
(School of Business,
Monash University Malaysia
siew.eu-gene@monash.edu)

Sang Jung Kang
(School of Information Technology,
Monash University Malaysia
sjkan2@student.monash.edu)

Abstract: We propose an effective technique to solving review-level sentiment classification problem by using sentence-level polarity correction. Our polarity correction technique takes into account the consistency of the polarities (positive and negative) of sentences within each product review before performing the actual machine learning task. While sentences with inconsistent polarities are removed, sentences with consistent polarities are used to learn state-of-the-art classifiers. The technique achieved better results on different types of products reviews and outperforms baseline models without the correction technique. Experimental results show an average of 82% F-measure on four different product review domains.

Key Words: Sentiment Analysis, Review-Level Classification, Polarity Correction, Data Mining, Machine Learning

1 Introduction

Sentiment classification has attracted a number of research studies in the past decade. The most prominent in the literature is Pang et al,[1] which employed supervised machine learning techniques to classify positive and negative sentiments in movie reviews. The significance of that work influenced the research

community and created different research directions within the field of sentiment analysis and opinion mining.[2, 3, 4] Practical benefits also emerged as a result of automatic recommendation of movies and products by using the sentiments expressed in the related review.[5, 3, 4] This is also applicable to business intelligence applications which rely on customers’ reviews to extract ‘satisfaction’ patterns that may improve profitability.[6, 7] While the number of reviews has continued to grow, and sentiments are expressed in a subtle manner, it is important to develop more effective sentiment classification techniques that can correctly classify sentiments despite natural language ambiguities, which include the use of irony.[8, 9, 10, 11]

In this work, we classify sentiments expressed on individual product types by learning a language model classifier. We focus on online product reviews which contain individual product domains and express explicit sentiment polarities. For example, it is quite common that the opinion expressed in reviews are targeted at the specific products on which the reviews are written.[2, 7] This enables the reviewer to express a substantial level of sentiments on the particular product alone without necessarily splitting the opinions between different products. Also, in a review, sentiments are likely to be expressed on specific aspects of the particular product.[12] For example, an iPad user may express positive sentiment about the ‘camera quality’ of the device but expresses negative sentiment about the ‘audio quality’ of the device. This provides useful and collaborative information on aspects of the product that need improvements.[13, 14, 15]

The application of sentiment classification is important to the ordinary users of opinion mining and sentiment analysis systems.[16, 2, 3] This is because the different categories of sentiments (e.g. positive and negative) represent the actual stances of humans on a particular target (e.g. a product). A product manufacturer for example, can have an overview of how many people ‘like’ and ‘dislike’ the product by using the number of *positive* and *negative* reviews. Similarly, sentiment classification has been quite useful in finance industries, especially for stock market prediction.[17, 18, 19]

Sentiment classification on product reviews can be challenging,[16, 3, 4] which is why it is still a very active area of research. More importantly, sentiments expressed in each product review sometimes include ambiguous and unexpected sentences, [20] and are often alternated between the two different positive and negative polarities. This causes inconsistencies in the sentiments expressed and consequentially leading to the mis-classification of the review document.[1, 16, 3] As such, the bag-of-words approach is not sufficient alone.[3, 4] We emphasize that most negative reviews contain positive sentences and often express negative sentiments by using just a few negative sentences.[21] We show an example as follows:

*I bought myself one of these and used it minimally and was
 happy (POSITIVE)*
I am using my old 15 year old Oster (NEGATIVE)
Also to my surprise it is doing a better job (POSITIVE)
Just not as pretty (NEGATIVE)
*I have KA stand mixer, hand blender, food processors large and
 small... (OBJECTIVE)*
Will buy other KA but not this again (NEGATIVE)

The above problem often degrades the accuracy of sentiment classifiers as many review documents get mis-classified to the opposite category. This is regarded as *false positives* and *false negatives* as the case may be.

While the above problem is non-trivial, we propose a polarity correction technique that extracts sentences with consistent polarities in a review. Our correction technique includes three separate steps. First, we perform training set correction by training a ‘naïve’ sentence-level polarity classifier to identify *false negatives* in both positive and negative categories. We then combine the *true positives* sentences and the *false negative* sentences of the two opposite categories to form a new training set for each category. Second, we propose a sentence-level polarity correction algorithm to identify consistent polarities in each review, while discarding sentences with inconsistent polarities. Finally, we learn different Machine Learning algorithms to perform the sentiment classification task.

The above steps were performed on four different Amazon product review domains and improved the accuracy of sentiment classification of the reviews over a baseline technique and give comparable performance with standard biagram, bag-of-words, and unigram techniques. In terms of F-measure, the technique achieve an average of 82% on all the product review domains.

The rest of this paper is organized as follows. We discuss related research work in Section 2. In Section 3, we propose the training set correction technique for sentiment classification task. Section 4 describes the sentence-level polarity correction technique and the corresponding algorithm. Our machine learning experiments and results are presented in Section 5. Finally, Section 6 presents conclusions and future work.

2 Related Work

Pang and Lee,[22] proposed a subjectivity summarization technique that is based on minimum cuts to classify sentiment polarities in IMDb movie reviews. The intuition is to identify and extract subjective portions of the review document using minimum cuts in graphs. The minimum cut approach takes into consideration, the pairwise proximity information via graph cuts that partitions sentences which are likely to be in the same class. For example, a strongly subjective sentence might have lexical dependencies on its preceding or next sentence. Thus Pang and Lee,[22] showed that minimum cuts in graph put such sentences in

the same class. In the end, the identified subjective portions as a result of the minimum graph cuts are then classified as either negative or positive polarity. This approach showed significant improvement from 82.8% to 86.4% with just 60% subjective portion of the documents.

In our work, we introduce additional steps by not only extracting subjective sentences. Instead, we extract subjective sentences with consistent sentiment polarities. We then discard other subjective sentences with inconsistent sentiment polarities that may contribute noise and reduce the performance of the sentiment classifier. Thus, contrary to Pang and Lee,[22] our work has the ability to effectively learn sentiments by identifying the likely subjective sentences with consistent sentiments. Again, we emphasize that some subjective sentences may not necessarily express sentiments towards the subject matter.[3, 4] Consider, for example, the following excerpt from a ‘positive-labelled’ movie review:

⁴real life, however, consists of long stretches of boredom with a few dramatic moments and characters who stand around, think thoughts and do nothing, or come and go before events are resolved. ²Spielberg gives us a visually spicy and historically accurate real life story. ³You will like it.’

In the above excerpt, sentence 1 is a *subjective* sentence which does not contribute to the sentiment on the movie. Explicit sentiments are expressed in sentence 2 and 3. We propose that discarding sentences such as sentence 1 from reviews is likely to improve the accuracy of a sentiment classifier.

Similarly, Wilson et al,[23] used instances of polar words to detect contextual polarity of phrases from the MPQA corpus. Each phrase detected is verified to be either *polar* or *non-polar* phrase by using the presence of opinionated words from a polarity lexicon. Polar phrases are then processed further to detect their respective contextual polarities which can then be used to train machine learning techniques. Identifying the polarity of phrase-level expression is a challenge in sentiment analysis.[3] Earlier in Section 1, we have illustrated some example sentences to that effect. For clarity, consider the sentence ‘*I am **not** saying the picture quality of the camera is **not good***’. In this sentence, the presence of the negation word ‘**not**’ does not represent ‘negative’ polarity of the sentence in context. In fact it emphasizes a ‘desired state’ that the ‘picture quality’ of the camera entity is ‘good’. However, without effective contextual polarity detection, such sentences could be easily classified as ‘negative’ by ordinary machine learning techniques. To this extent, Wilson et al,[23] performed manual annotation of contextual polarities in the MPQA corpus to train a classifier with a combination of ten features resulting to 65.7% accuracy giving room for more improvement.

Choi and Cardie,[24] proposed a *compositional semantics* approach to learn the polarity of sentiments from the sub-sentential level of opinionated expres-

sions. The compositional semantic approach breaks the lexical constituents of an expression into different semantic components. Thus, the work used *content word negators* (e.g. sceptic, disbelief) to identify the sentiment polarities from the different semantic components of the expression. Content word negators are *negation* words other than *function* words such as *not*, *but*, *never* and so on. Identified sentiment polarities are then combined using a set of heuristic rules to form an overall sentiment polarity feature which can then be used to train machine learning techniques. Interestingly, on the Multi-Perspective Question Answering (MPQA) corpus created by Wiebe et al.[25] this combination yielded a performance of 90.7% over the 89.1% performance of ordinary classifier (e.g. using bag-of-words).

The performance achieved by Choi and Cardie,[24] is understandable given that the MPQA corpus contains well ‘structured’ news articles which are mostly well written on certain topics. Moreover, sentences or expressions which are contained in news articles are most likely to express *sequential sentiments* for a reasonable classification performance.[17, 26, 27] For example, it is more likely that a negative news ‘event’ such as ‘*Disaster unfolds as Tsunami rocks Japan*’ will attract ‘persistent’ negative expressions and sentiments in news articles. In contrast, sentiment classification on product reviews is more challenging as there is often inconsistent or mixed sentiment polarities in the reviews. We have illustrated an example to that effect in Section 1. It would be interesting to know the performance of the heuristics used by Choi and Cardie,[24] on standard product review datasets such as Amazon online product review datasets. A detailed review of other sentiment classification techniques on review documents is provided in Tang et al.[28]

Our main contribution to the sentiment classification task is to do training set correction and further detect *inter-sentence* polarity consistency that could improve a sentiment classifier. That is, given a review of n -sentences, we try to understand how the sentiment polarity varies from sentence 1 to sentence n . We hypothesize that detecting *consistent sentiment patterns* in reviews could improve a sentiment classifier without further sophisticated natural language techniques (e.g. using compositional semantics or linguistic dependencies).[17]

More importantly, we believe every sentence in the review may not necessarily contribute to the classification of the review to the appropriate class.[22] We say that certain *sequential sentences with consistent sentiment polarities* could be sufficient to represent and distinguish between the sentiment classes of a review. Representative features have been argued to be the key to effective classification technique.[29, 30] We emphasize that our approach is promising and can be easily integrated by any sentiment classification system regardless of the sentiment detection technique employed.

3 Training Set Correction

Training set polarity correction has been largely ignored in sentiment classification tasks.[6] Earlier, we emphasized that a review document could contain both positive and negative sentences. Moreover, since reviewers often express sentiments on different aspects of products, it is probable that some aspects of the products will receive positive sentiments while others get negative sentiments.[3] In a negative-labeled product review for example, it is more likely that negative sentiments will be expressed within the first few portion of the review and then followed by positive sentiments in the later portion of the review on some of the aspects of the product that gave some satisfactions.[5, 3] This could be because reviewers tend to emphasize on the negative aspects of a product than the positive aspects, and in some cases, both polarities are expressed alternately, which we will discuss in Section 4. Thus, using such mixed sentiments in each category, for training a machine learning algorithm will only result to bias and reduce the accuracy of the classifier.[22, 31]

As such, we propose a promising approach to reduce the bias in the training set by first learning a ‘naïve’ sentence-level classifier on all sentences from both the positive and negative categories. A ‘naïve’ classifier could be any classifier trained with surface-level features (e.g. unigram or bag-of-words),[22, 32] without necessarily performing sophisticated features engineering since the final sentiment classifier will be constructed with more fine-grained features. [33] For example, one could learn the popular Naïve Bayes classifier with only unigram features.[22, 34, 35] It is also possible to use a more complexly constructed classifier at the expense of efficiency. Having said that, the ‘naïve’ classifier is then used to also test the same sentences from both the positive and negative categories. The idea is to identify *positive-labelled sentences* that will be classified as negative and *negative-labelled sentences* that will be classified as positive. Having identified this, it is therefore imperative to correct the training set by combining the wrongly classified sentences to their original respective categories. That is, positive-labelled sentences that are classified as negative should be combined with the original negative sentences (in the negative category) and negative-labelled sentences that are classified as positive should be combined with the original positive sentences (in the positive category).

While this technique may result to a meta classification,[36] we propose to include the technique as part of the training process of the final sentiment classifier. In addition, in order to minimize wrongly classified sentences, we implement the ‘naïve’ classifier to maximize the *Joint-Log-Probability* score of a given sentence belonging to either of positive or negative categories. This is because most ordinary classifiers maximize the conditional probability over all categories, which is at the expense of better accuracy.[37] We compute the *Joint-Log-Probability* as follows:

$$P(S, C) = \log_2 P(S|C) + \log_2 P(C) \quad (1)$$

$$P_c = \operatorname{argmax}_{c \in C} P(S, C) \quad (2)$$

where $P(S, C)$ is the probability of a sentence given a class, P_c is the probability of the sentence belonging to either a ‘positive’ category c or a ‘negative’ category c and $P(C)$ is a multivariate distribution on the positive and negative categories.

4 Sentence-Level Polarity Correction

Following the training set correction in Section 3, we propose the sentence-level polarity correction to further reduce mis-classification in both ‘training’ and ‘testing’ sets. More importantly, because the bag-of-words approach has seldom improve the accuracy of a sentiment classifier,[3, 4] a sentence-level approach could give better improvement since most sentiments are expressed at sentence-level anyway.[38] However, we have indicated in Section 3 that many review documents have the tendency to contain both positive and negative sentences, regardless of their individual categories (i.e. positive or negative). While the consistent sentence polarities of both categories might be helpful to the classification task, it would be better to remove sentences with *outlier* polarities that cause inconsistencies by using a polarity correction approach.[39, 40, 3] Note that we have motivated the inconsistency problem with an example in Section 1.

The idea of the sentence-level polarity correction is to remove inconsistent sentence polarities from each review. We observed that sentences with inconsistent polarity deviate from the previous consistent polarity. More often than not, the polarities of sentences in a given review are expressed consistently except for some outliers polarities.[1, 40, 3] As such, a given polarity is expressed consistently over a number of sentences and at a certain point deviate to the other polarity, and continues over a number of sentences alternately. Figure 1 shows an illustration depicting a possible review with consistent polarities and inconsistent polarities (or outlier polarities).

Given a 10-sentence review, a reviewer has expressed negative sentiments with the first three sentences. This is followed by a single positive sentence on line 4. Lines 5 to 7 consist of another three negative sentences. Lines 8 to 9 expressed positive sentences. Finally, line 10 concluded with a negative sentence. Thus, we regard line 4 (positive sentence) and line 10 (negative sentence) as outlier polarities because there is no subsequent exact polarity after each of them. Our polarity correction algorithm removes such outliers, leaving only the consistent polarities. It is to be noted that at this stage, the algorithm is independent of a particular sentiment category (i.e. positive or negative). We consider exact

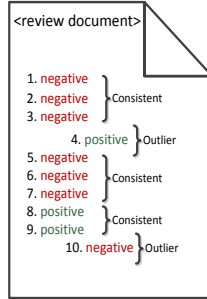


Figure 1: An illustration of a review document with outliers, consistent, and inconsistent polarities.

subsequent polarities - either positive or negative - since a review is likely to contain both polarities as discussed earlier. Our intuition is that sentences with consistent polarity could better represent the overall sentiment expressed in a review document by providing a wider margin between the categories of the major consistent sentiment polarities.[17, 21] Note that this technique is different from *intra-sentence* polarity detection as studied in Li et al.[40] An additional thing we did was to performed negation tagging by tagging 1 to 3 words after a negation word in each sentence. In contrast to our baseline, the negation tagging showed some improvements in our correction technique.

Thus, given a review document with n -number of sentences S_1, \dots, S_n , we classify each sentence with the ‘naïve’ classifier and compare the polarity Φ_s of the first sentence with the polarity $\Phi_{s_{n+1}}$ of the next sentence until s_{n-1} . Where Φ_s is the starting polarity, the polarity of the subsequent sentence $\Phi_{s_{n+1}}$ is compared with the polarity of the *prior* sentence $\Phi_{\lambda_{s_{n+1}}}$. When $\Phi_{\lambda_{s_{n+1}}}$ equals $\Phi_{s_{n+1}}$, the sentence is stored into the consistent category, otherwise, the sentence is considered outlier. Note that we set a *consistency* threshold by specifying a parameter θ , which indicates the minimum number of subsequent and the same sentence polarities that must be considered consistent. As such, consistent sentence polarities that are lower than the θ value are ignored.

In our experiment, we set $\theta = 2$ to simulate the default case. Our empirical observation shows that $\theta = 2$ sufficiently captures consistent polarities for a sparse review document containing as low as 7 sentences. Figure 2 shows how consistent polarities are extracted with different threshold θ , where $\theta = 2$ retrieves sentences n_3 to n_7 and $\theta = 3$ retrieves only sentences n_5 to n_7 .

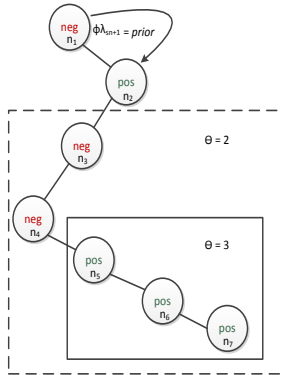


Figure 2: Extracting consistent polarities with different θ thresholds.

5 Experiment and Results

We performed several experiments with our correction technique and compare between the performance on popular state-of-the-art classifiers with and without our polarity correction techniques. The classifiers comprises of the Sequential Minimum Optimization (SMO) variant of Support Vector Machines (SVM),[41] and Naïve Bayes (NB) classifier.[42] We used SVM and NB on the WEKA machine learning platform,[43] with *bag-of-words*, *unigram*, and word *bigram* features. We did not include word *trigram* features as both word unigram and word bigram features have been studied to improved sentiment classification tasks.[1, 22, 3] We conducted **80%-20%** performance evaluation for comparison with the baselines on each dataset domain.

For selecting the best parameters for the baseline algorithms, we performed hyperparameters search using Auto-Weka,[44] with cross-validation and the Sequential Model-based Algorithm Configuration (SMAC) optimization algorithm, which is an Bayesian optimization method proposed as part of Auto-Weka.[44] We performed the search by using the unigram features on the training set of each domain. This is because unigram features have shown robust performance in sentiment analysis.[1, 22, 3]

5.1 Dataset and Baseline

Our dataset is the multi-domain sentiment dataset constructed by Blitzer et al.[5] The dataset was first used in year 2007 and consists of Amazon online

Model	Hyperparameters
SVM-beauty	-C 1.1989425641153333 -N 0 -K "NormalizedPolyKernel -E 1.6144079568156302 -L"
SVM-books	-C 1.2918141993816825 -N 2 -K "NormalizedPolyKernel -E 2.78637472738497"
SVM-kitchen	-C 1.2929645940353218 -N 2 -K "Puk -S 9.028189222927269 -O 0.9952824838773323"
SVM-software	-C 1.1471978195519354 -N 2 -M -K "NormalizedPolyKernel -E 1.7177045231155679 -L"
NB-all-domains	-K (Kernel Estimator)

Table 1: Auto-Weka hyperparameters settings for SVM and NB on product domains with unigram features.

product reviews from four different types of product domains¹, which includes, *beauty products*, *books*, *software*, and *kitchen*. Each product domain has **1000** positive reviews and **1000** negatives reviews, which were identified based on the customers’ star ratings according to Blitzer et al.[5] For each domain, we separated **800** documents per category as *training set* and used the remaining **200** documents as *unseen testing set*. We extracted the review text and performed sentence boundary identification by optimizing the output of the *MedlineSentenceModel* available as part of the LingPipe library.²

As our baseline, we implemented a sentence-level sentiment classifier using a technique similar to Pang and Lee,[22] on the same dataset but without our correction technique. The baseline technique has worked very well in most sentiment classification tasks. The baseline work removes objective sentences from the training and testing documents by using an automatic *subjectivity detector* component which uses subjective sentences only for sentence-level classification.

5.2 Evaluation

We used three evaluation metrics comprising of *precision*, *recall*, and *F-Measure* or *F-1*. The *precision* is calculated as $TP/(TP + FP)$, *recall* as $TP/(TP + FN)$, and *F-Measure* as $(2 * precision * recall)/(precision + recall)$. Note that TP, TN, FP, and FN are defined as true positives, true negatives, false positives, and false negatives, respectively. All results are based on 95% Confidence Interval.

5.3 Results and Discussion

We present the results in Tables 2 - 5, where *Model* is the type of classifier, *Pr.* is the precision, *Rc.* is the recall, and *F-1* is the F-measure, respectively. We identify the models with our correction technique with ‘cor’ after the model names. For

¹ <http://www.cs.jhu.edu/~mdredze/datasets/sentiment/>

² <http://alias-i.com/lingpipe/docs/api/com/aliasi/sentences/MedlineSentenceModel.html>

Model	Pr.	Rc.	F-1
SVM-Bigram-cor	0.83	0.83	0.83
SVM-BOWS-cor	0.83	0.83	0.83
SVM-Unigram-cor	0.84	0.85	0.84*
SVM-Bigram	0.83	0.83	0.83
SVM-BOWS	0.85	0.85	0.85†
SVM-Unigram	0.83	0.83	0.83
SVM-Baseline	0.76	0.76	0.76
NB-Bigram-cor	0.83	0.79	0.80
NB-BOWS-cor	0.83	0.81	0.81*
NB-Unigram-cor	0.79	0.74	0.75
NB-Bigram	0.86	0.86	0.86†
NB-BOWS	0.83	0.83	0.83
NB-Unigram	0.86	0.85	0.86
NB-Baseline	0.75	0.75	0.75

Table 2: Performance of unseen test sets on Beauty Reviews

Model	Pr.	Rc.	F-1
SVM-Bigram-cor	0.83	0.82	0.82
SVM-BOWS-cor	0.84	0.84	0.84
SVM-Unigram-cor	0.82	0.82	0.82
SVM-Bigram	0.84	0.83	0.83
SVM-BOWS	0.82	0.82	0.82
SVM-Unigram	0.85	0.85	0.85†
SVM-Baseline	0.70	0.70	0.70
NB-Bigram-cor	0.84	0.84	0.84
NB-BOWS-cor	0.80	0.79	0.79
NB-Unigram-cor	0.82	0.82	0.82*
NB-Bigram	0.83	0.79	0.79
NB-BOWS	0.77	0.77	0.76
NB-Unigram	0.79	0.77	0.77
NB-Baseline	0.72	0.72	0.72

Table 4: Performance of unseen test sets on Software reviews

Model	Pr.	Rc.	F-1
SVM-Bigram-cor	0.77	0.77	0.77
SVM-BOWS-cor	0.81	0.81	0.81*
SVM-Unigram-cor	0.75	0.75	0.75
SVM-Bigram	0.76	0.75	0.76
SVM-BOWS	0.78	0.78	0.78
SVM-Unigram	0.78	0.77	0.77
SVM-Baseline	0.68	0.68	0.68
NB-Bigram-cor	0.78	0.74	0.74
NB-BOWS-cor	0.84	0.81	0.81*
NB-Unigram-cor	0.79	0.77	0.77
NB-Bigram	0.78	0.77	0.76
NB-BOWS	0.56	0.55	0.53
NB-Unigram	0.69	0.67	0.66
NB-Baseline	0.69	0.67	0.64

Table 3: Performance of unseen test sets on Books reviews

Model	Pr.	Rc.	F-1
SVM-Bigram-cor	0.79	0.79	0.79
SVM-BOWS-cor	0.85	0.85	0.85*
SVM-Unigram-cor	0.79	0.79	0.79
SVM-Bigram	0.81	0.79	0.79
SVM-BOWS	0.81	0.8	0.80
SVM-Unigram	0.79	0.79	0.79
SVM-Baseline	0.74	0.74	0.74
NB-Bigram-cor	0.85	0.82	0.82
NB-BOWS-cor	0.82	0.82	0.82
NB-Unigram-cor	0.82	0.82	0.82
NB-Bigram	0.84	0.84	0.84†
NB-BOWS	0.77	0.76	0.76
NB-Unigram	0.82	0.82	0.82
NB-Baseline	0.72	0.72	0.72

Table 5: Performance of unseen test sets on Kitchen reviews

example ‘SVM-Unigram-Cor’ depicts a model using SVM with unigram features and our correction techniques. Standard models are identified by the algorithm name and the feature used. Baseline models are identified with ‘Baseline’. In addition, we identify our best performing model above the baseline with (*) and comparable performance with standard models is identified with (†).

We see that the model with our correction techniques outperformed the baseline model without the correction techniques on all domains. Other than the baseline model, our technique show comparable performance with the standard bigram, bag-of-words, and unigram models. Not surprisingly, SVM performed better than NB in most cases with bag-of-words and unigram features. On the other hand, NB performed better than SVM with bigram features. The improvement on the baseline technique and the comparable performance on the standard models show the importance of our polarity correction techniques as

applicable to sentiment classification. It also emphasizes the fact that using the unseen test sets without sentence-level polarity corrections is likely to lead to mis-classification as a result of inconsistent polarities within each review. Perhaps, it could be beneficial to consider the integration of our polarity correction techniques into an independent sentiment classifier for more accurate sentiment classification.

The limitation of our polarity correction techniques, however, could be in the construction and the performance of the initial ‘naïve’ classifier for performing both the training set and the sentence-level polarity corrections. Also, the classifier needed to be trained on each review domain. At the same time, we emphasize that a moderate classifier - taking a NB classifier as an example - trained with the standard bag-of-word features, gives an average of approximately 72% F-measure across all domains as observed in our results. Therefore, we believe that the process is likely to have a minimal or negligible effect on the resulting sentiment classifier. As such, in favor of a more efficient classification task, especially on very large datasets, we do not recommend sophisticated classifiers for the initial correction processes. We also like to emphasize that any reasonable sentence-level polarity identification technique,[3] used in place of the ‘naïve’ classifier in the correction processes, is likely to work just fine and give improved results for the overall sentiment classification task.

6 Conclusions

In this work, we have proposed a training set and sentence-level polarity correction for the sentiment classification task on review documents. We performed experiments on different Amazon product review domains and show that a sentiment classifier with training set and sentence-level polarity corrections, showed improved performance and outperformed a state-of-the-art sentiment classification baseline on all the review domains. Our correction techniques first remove polarity bias from the training set and then inconsistent sentence-level polarities from both training and testing sets. Given the difficulty of the sentiment classification task [3], we believe that the improvement shown by the correction technique is promising and could lead to building a more accurate sentiment classifier.

In the future, we will integrate the training and sentence-level polarity correction techniques as part of an independent sentiment detection algorithm and perform larger scale experiment on large datasets such as the SNAP Web Data: Amazon reviews dataset³, which was prepared by McAuley and Leskovec.[45]

³ <http://snap.stanford.edu/data/web-Amazon.html>

References

1. B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing (EMNLP)*, pp. 79–86, Association for Computational Linguistics, 2002.
2. O. Vechtomova, “Facet-based opinion retrieval from blogs,” *Information Processing & Management*, vol. 46, no. 1, pp. 71–88, 2010.
3. B. Liu, “Sentiment analysis and opinion mining,” *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.
4. E. Cambria, B. Schuller, Y. Xia, and C. Havasi, “New avenues in opinion mining and sentiment analysis,” *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, 2013.
5. J. Blitzer, M. Dredze, and F. Pereira, “Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification,” (Association of Computational Linguistics (ACL)), 2007.
6. E. Martínez-Cámara, M. T. Martín-Valdivia, L. A. Ureña-López, and A. R. Montejo-Ráez, “Sentiment analysis in twitter,” *Natural Language Engineering*, vol. 20, no. 01, pp. 1–28, 2014.
7. D. Vilares, M. A. Alonso, and C. Gómez-Rodríguez, “A syntactic approach for opinion mining on spanish reviews,” *Natural Language Engineering*, pp. 139–163, 2015.
8. J. Mendel, L. Zadeh, E. Trillas, R. Yager, J. Lawry, H. Hagra, and S. Guadarrama, “What computing with words means to me [discussion forum],” *Computational Intelligence Magazine, IEEE*, vol. 5, no. 1, pp. 20–26, 2010.
9. F. Keshkar and D. Inkpen, “A hierarchical approach to mood classification in blogs,” *Natural Language Engineering*, vol. 18, no. 01, pp. 61–81, 2012.
10. A. Reyes and P. Rosso, “On the difficulty of automatically detecting irony: beyond a simple case of negation,” *Knowledge and Information Systems*, vol. 40, no. 3, pp. 595–614, 2014.
11. M. Melero, M. Costa-Jussà, P. Lambert, and M. Quixal, “Selection of correction candidates for the normalization of spanish user-generated content,” *Natural Language Engineering*, pp. 1–27, 2014.
12. Y. Jo and A. H. Oh, “Aspect and sentiment unification model for online review analysis,” in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 815–824, ACM, 2011.
13. A. Tsai, R. Tsai, and J. Hsu, “Building a concept-level sentiment dictionary based on commonsense knowledge,” *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 22–30, 2013.
14. S. Poria, A. Gelbukh, A. Hussain, D. Das, and S. Bandyopadhyay, “Enhanced sentiment with affective labels for concept-based opinion mining,” *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 31–38, 2013.
15. N. T. Roman, P. Piwek, A. M. B. R. Carvalho, and A. R. Alvares, “Sentiment and behaviour annotation in a corpus of dialogue summaries,” *Journal of Universal Computer Science*, vol. 21, no. 4, pp. 561–586, 2015.
16. B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.
17. A. Devitt and K. Ahmad, “Sentiment polarity identification in financial news: A cohesion-based approach,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 984–991, ACL, 2007.
18. A. Brabazon, M. O’Neill, and I. Dempsey, “An introduction to evolutionary computation in finance,” *Computational Intelligence Magazine, IEEE*, vol. 3, no. 4, pp. 42–55, 2008.

19. E. J. Fortuny, T. D. Smedt, D. Martens, and W. Daelemans, "Evaluating and understanding text-based stock price prediction models," *Information Processing & Management*, vol. 50, no. 2, pp. 426 – 441, 2014.
20. D. Li, A. Laurent, P. Poncelet, and M. Roche, "Extraction of unexpected sentences: A sentiment classification assessed approach," *Intelligent Data Analysis*, vol. 14, no. 1, p. 31, 2010.
21. L. Jia, C. Yu, and W. Meng, "The effect of negation on sentiment analysis and retrieval effectiveness," in *Proceeding of the 18th ACM conference on Information and knowledge management*, (Hong Kong, China), pp. 1827–1830, ACM, 2009.
22. B. Pang and L. Lee, "A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, (Barcelona, Spain), p. 271, Association for Computational Linguistics, 2004.
23. T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, (Vancouver, British Columbia, Canada), pp. 347–354, Association for Computational Linguistics, 2005.
24. Y. Choi and C. Cardie, "Learning with compositional semantics as structural inference for subsentential sentiment analysis," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (Honolulu, Hawaii), pp. 793–801, Association for Computational Linguistics, 2008.
25. J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language Resources and Evaluation*, vol. 39, no. 2/3, pp. 165–210, 2005.
26. M. Bautin, C. B. Ward, A. Patil, and S. S. Skiena, "Access: news and blog analysis for the social sciences," in *Proceedings of the 19th international conference on World wide web*, (Raleigh, North Carolina, USA), pp. 1229–1232, ACM, 2010.
27. Y. Lee, H.-y. Jung, W. Song, and J.-H. Lee, "Mining the blogosphere for top news stories identification," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, (Geneva, Switzerland), pp. 395–402, ACM, 2010.
28. H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10760–10773, 2009.
29. I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]," *Computational Intelligence Magazine, IEEE*, vol. 5, no. 4, pp. 13–18, 2010.
30. Y. He and D. Zhou, "Self-training from labeled features for sentiment analysis," *Information Processing & Management*, vol. 47, no. 4, pp. 606–616, 2011.
31. P. H. Calais Guerra, A. Veloso, W. Meira Jr, and V. Almeida, "From bias to opinion: a transfer-learning approach to real-time sentiment analysis," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 150–158, ACM, 2011.
32. H. Cui, V. Mittal, and M. Datar, "Comparative experiments on sentiment classification for online product reviews," (American Association for Artificial Intelligence (AAAI)), 2006.
33. S. Maldonado and C. Montecinos, "Robust classification of imbalanced data using one-class and two-class svm-based multiclassifiers," *Intelligent Data Analysis*, vol. 18, no. 1, pp. 95–112, 2014.
34. S. Tan, X. Cheng, Y. Wang, and H. Xu, "Adapting naive bayes to domain adaptation for sentiment analysis," in *Advances in Information Retrieval*, pp. 337–349, Springer, 2009.
35. Q. Ye, Z. Zhang, and R. Law, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6527–6535, 2009.

36. R. Xia, C. Zong, X. Hu, and E. Cambria, "Feature ensemble plus sample selection: A comprehensive approach to domain adaptation for sentiment classification," *IEEE Intelligent Systems*, vol. 28, no. 3, pp. 10–18, 2013.
37. D. Lowd and P. Domingos, "Naive bayes models for probability estimation," in *Proceedings of the 22nd international conference on Machine learning*, ICML '05, (New York, NY, USA), pp. 529–536, ACM, 2005.
38. L. Tan, J. Na, Y. Theng, and K. Chang, "Sentence-level sentiment polarity classification using a linguistic approach," *Digital Libraries: For Cultural Heritage, Knowledge Dissemination, and Future Creation*, pp. 77–87, 2011.
39. T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis," *Computational Linguistics*, vol. 35, no. 3, pp. 399–433, 2009.
40. S. Li, S. Y. M. Lee, Y. Chen, C.-R. Huang, and G. Zhou, "Sentiment classification and polarity shifting," in *Proceedings of the 23rd International Conference on Computational Linguistics*, (Beijing, China), pp. 635–643, Association for Computational Linguistics, 2010.
41. J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Tech. Rep. MSR-TR-98-14, Microsoft Research, 1998.
42. I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, pp. 41–46, 2001.
43. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
44. C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-weka: Combined selection and hyperparameter optimization of classification algorithms," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 847–855, ACM, 2013.
45. J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172, ACM, 2013.